Early Area and Power Estimation Model For Rapid System Level Design and Design Space Exploration

Abhishek Narayan TRIPATHI[1], *Arvind RAJAWAT*[2]

[1] Department of Electronics and Communication Engineering, National Institute of Technology,
G.E. Road, Raipur, 492010 Chhattisgarh, India
[2]Department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology,
Link Road Number 3, Near Kali Mata Mandir, Bhopal, 462003 Madhya Pradesh, India

antripathi.etc@nitrr.ac.in, rajawata@manit.ac.in

DOI: 10.15598/aeee.v20i1.4229

Article history: Received May 20, 2021; Revised Nov 08, 2021; Accepted Nov 24, 2021; Published Mar 31, 2022.

Abstract. *Power and area estimation in the early stage of designing is very critical for a system. This paper presents the neural network-based early area and power estimation model. The flow starts with the training of the neural network model from the selected behavioral level parameters, which imposes to provide accurate estimations. The model accuracy is validated against ITC99 benchmark programs. The run-times are faster than the synthesis run-times. For the ASIC-based designs, the proposed model took 5 seconds, while Synopsys Design Compiler took 5 minutes. In terms of timing, the estimation speed is more than the order of magnitude faster than the conventional synthesis-based approach. The modeling methodology provides a better, accurate, and fast area and power estimations, at an early stage of the Very-Large-Scale Integration (VLSI) design. In addition, the model eliminates the need for synthesis-based exploration and provides the design picking before synthesis.*

Keywords

Area estimation, design space exploration, neural network, power estimation, VLSI.

1. Introduction

Raising the abstraction level to the system level is the most important countermeasure adopted to handle the increasing complexity of System on Chips (SoCs). Decisions taken at this early stage of design cycle level have the greatest impact on the final design in terms of the design metrics such as performance, power, area, etc [1]. However, taking decisions at this level is very difficult, since the design space is extremely wide. Efficient system-level estimation methods are therefore necessary for designspace exploration. The designs are getting complex to incorporate more functions, thereby increasing the area and power dissipation. Therefore, a quick and accurate estimation of power and area characteristics is of a paramount importance to guide the decision-making process.

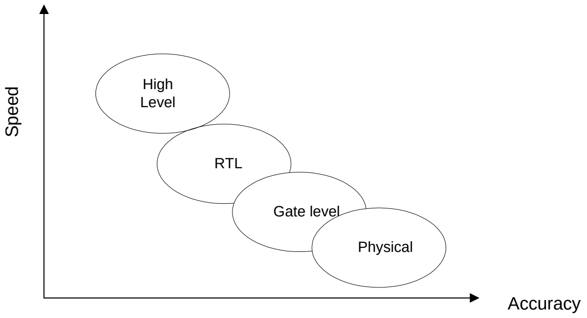


Fig. 1: Speed accuracy trade-off.

2. Literature Review

For the area estimation, some techniques are tailored for certain partitioning schemes [2] and [3]. Such schemes are suitable for iterative partitioning algorithms where the area of the hardware part is updated after adding (removing) any component to (from) the hardware side. Other techniques estimate the hardware area independently of the partitioning process. Most of the published work performs a transformation step to express the input description into an Intermediate Representation (IR) such as Trimaran IR [4], Control Data Flow Graph (CDFG) [5], and VHDL AST [6], and then, apply the estimation process on the intermediate format. In [7], a technique was proposed to estimate the Field Programmable Gate Array (FPGA) area utilization of the data flow graphs (DFGs) from applications. The technique using DFGs can again be divided into different categories and a formula can be developed for each category to estimate the area. In [8], an area estimation approach was presented for the look-up-table based FPGAs that take into account not only gate area and delay, but also the wiring effects. Similarly, in [9] an area model was presented which is based on transforming the given multi-output Boolean function description into an equivalent single-output function. The primary issue in both the work is the model complexity. In [10], a parameterized macromodel was presented which is derived by actual synthesis of Register Transfer Language (RTL) operators using back- end logic synthesis and place-and-route tools. However, the time to get the final result was more significant.

Similarly, Significant research has been done to provide a low power solution [11], [12] and [13], and models have been developed for energy and power estimation at the highlevel of abstraction in embedded system design. In [14], simulation-based power estimation tool, Powersim, was presented for SystemC designs at system-level. This model uses the energy of each operation by simulating the designs; nevertheless, the reported relative error is more than 15 %.

By identifying computing resources of heterogeneous SoC, neural network-based system level power estimation was presented in [15]. The maximum error of 31 %, using a linear model, and 4.78 % error is reported using a non-linear model.

[16] presents an equation based resource utilization model for automatically generated DFT soft core IP. However, the estimation error is about 6 %.

In [17], input data based simulator for power estimation has been presented. There is a high mismatch of the power estimation result of the simulator to the VPR cad tool. The reported estimation error varies from 35 to 82 %. The leakage power for ASIC designs was estimated in [18]. The linear regression model with a maximum error of 12 % has been reported for the Hardware Description Language (HDL) of the circuits.

In [19], an approach to estimate the amount of the required "on" capacitance of each decoupling capacitors at runtime to achieve runtime decoupling capacitors modulation in multi-core chips was proposed. This work was motivated by the characteristic of the power profiling of circuit blocks in a processor chip.

Power Inference using Machine Learning (PRIMAL), a machine learning based power estimation framework at RTL level for ASIC based designs was proposed in [20]. However, to speed-up the design cycle it can be shifted at the higher level of abstraction.

To summarize there is a need for the fast and efficient area and power estimation model for a given high-level application.

The remainder of this paper is organized as follows. Section 3. describes the modeling methodology. Section 4. discusses the results and the validation. Finally, Sec. 5. concludes the paper.

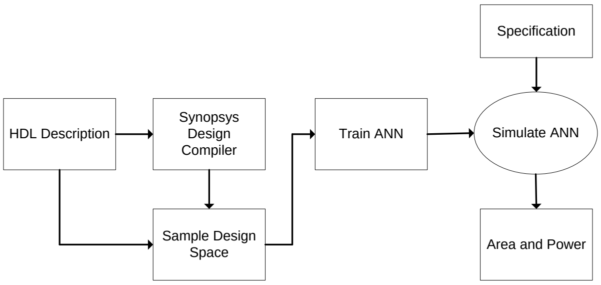


Fig. 2: Estimation flow.

3. Methodology

The basic component of estimation process is given below in the Fig. 2. Initially, the HDL description is given to the Synopsis Design Compiler [21] to get the area and power details of the designs. The next block, sample design space, contains samples of the power and area along with number of statement given in HDL description. The profiling of the applications to get the sample space is discussed in the next sub-section.This information is given as input to the artificial neural network block [22] for training of the network. The output unit of this supervised Artificial Neural Network (ANN) consist of the area and power.

3.1. Model Parameters

The primary step was the profiling of the training applications which involves the identification of the model parameters. Training set generation and training of the neural network. The initial step for estimation is generation of good number of samples to train the neural network. For the generation of training set the behavioral level description was sampled in terms of:

- Type of statements.

- Type of bits.
- Clock frequency.

1) Types of Statements

In HDL Description there are three types of statements.

- Assignment Statements: The assignment statements are those in which computed value (either constant or computed from expression) assigned to the other signals or case blocks. The assignment statements are responsible for net switching power only. But the area depends on number of cells.
- Control Statements: Control assignments are the type of statements in which a certain if-else conditions or case statements are there.
- Computational Statements: Computational statements are those in which arithmetic and logic expressions are analyzed. These statements are responsible for cell internal power only. The area varies according to the type of arithmetic and logical expression.

If S_a, S_c, and S_{comp} represents the three statements and n represents the number of input-output bits then the total power and area can be represented as:

$$P_t = f_1(S_a, S_c, S_{comp}, n), \tag{1}$$

$$A_t = f_2(S_a, S_c, S_{comp}, n). \tag{2}$$

The area would be equivalent to the implementation units of the statements. The bit-size n is introduced to signify the difference between the bit-size of the assignments statements and computation statements. As different bit-size will realize a different hardware and thus, cause a change in the power and area.

2) Types of Bits

Type of bits refers to number of bits required to perform that operation. In HDL description there are instructions which require different number of bits to perform assignment or computational operation. On increasing the number of bits the power is also increases proportionally.

$$y \leq a + b. \tag{3}$$

For example, in Eq. (1), suppose a and b are two 1 bit numbers and the power for the presented operation would be low as compared to the two 2 bit numbers. As for 2 bit operation it requires 2 bit adder made from two 1 bit adder. Consequently, it increases the area and power.

3) Clock Frequency

As from the expression of power:

$$P = \alpha(CV_{dd}^2)f. \tag{4}$$

Where α is the activity factor, C is the load capacitance, V_{dd} is the supply voltage, and f is the clock frequency. f is directly proportional to the power therefore Clock frequency of an operation is also a parameter to consider into accounts for power.

Figure 3 is the proposed model for area and power estimation which contains assignment statement, control statements, computational statements, type of bits, and clock frequency as inputs, and area and power as outputs. The default activity factor of 50 % has been used. The operating frequency, capacitive load C, and power supply were 300 MHz, 13 fF, and 1.2 V, respectively.

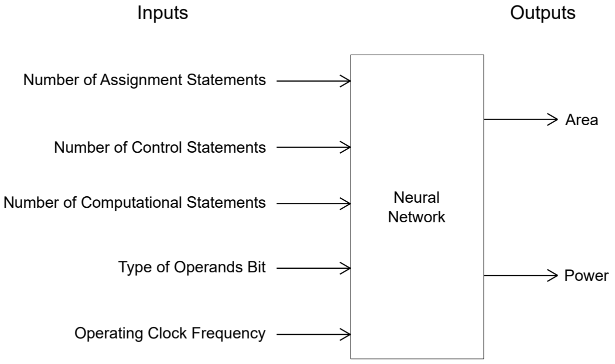

Inputs
Outputs
Number of Assignment Statements
Number of Control Statements
Number of Computational Statements
Type of Operands Bit
Operating Clock Frequency
Neural Network
Area
Power

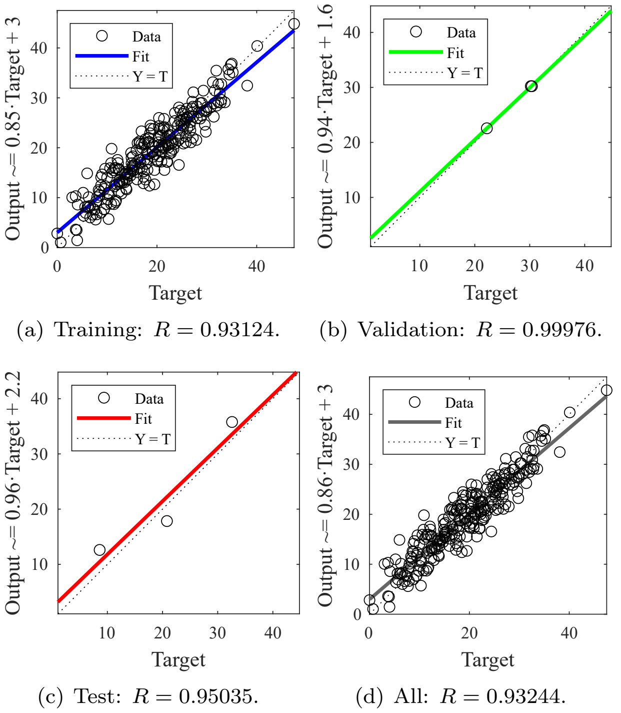
Fig. 3: Proposed model for estimation.

The output of the proposed model having 5 inputs is given with the following equations:

$$\frac{P}{A_n} = \text{tansig}\left(\sum_{n=1}^{n=5} w_{nf} I_n + b_{Ff}\right), \tag{5}$$

$$Output = \text{tansig}\left(\sum_{f=1}^{f=5} \frac{P}{A_n} w_{FfO} + b_{OF}\right). \tag{6}$$

In the above expression I_n are the inputs, w_{nf} are the weights from the inputs to the hidden layer neurons, b_{Ff} are the biases to the hidden layer neurons, w_{FfO} are the weights of hidden layer neurons to the output layer neuron, and b_{OF} is the bias to the output neuron. The training of ANN is carried out with *trainlm* learning algorithm. *trainlm* supports train6ing with validation and test vectors. A total of 31 applications were used for the training. Out of which 15 % applications were used for validation and 15 % were used for the testing during the training phase. After that, the trained model was validated for the different benchmark applications Validation vectors have been used to stop over-fitting. Test vectors have been used as a further check that the network is generalizing well. The regression plot for training, validation, and testing of dataset is shown in Fig. 4. The regression value for the training, validation, and test set is very close to 1, which shows that the proposed model is performing good for the new set of test data as well.



(a) Training: $R = 0.93124$.

(b) Validation: $R = 0.99976$.

(c) Test: $R = 0.95035$.

(d) All: $R = 0.93244$.

Fig. 4: Regression plot for training, validation and testing dataset.

4. Results and Discussion

In this section, we have validated the estimated area and power against the area and power obtained from the commercial tool using applications which are different from the training. The results obtained from the Synopsys Design Compiler are compared with the estimated results obtained from the proposed method.

The proposed estimation model is applied on the nine ITC benchmark [23] applications.

$$error_i = \left(\frac{e_i - p_i}{p_i}\right) \cdot 100. \tag{7}$$

In the above expression, $error_i$ is the percentage error in the estimate, e_i is the estimated area and power obtained from the model, and p_i is the area and power obtained from the Synopsys Design Compiler, for an application i.

The power estimation comparison with classical method is shown in Fig. 5. The relative error percentage in the power estimation is shown in Fig. 6.

The relative errors of 0.19 % to 9.33 % and 0.19 % to 7.43 % are observed for power and area estimation, respectively.

Tab. 1: Comparison of Estimated power and Synthesis power for ITC99 benchmarks.

Benchmarks [23]	Synthesis power (mW)	Estimated power (mW)	Relative error
B1	228.3053	206.989	9.33
B2	163.0591	168.2312	3.17
B3	1403.591	1373.2681	2.16
B4	2188.65	2184.382	0.19
B6	430.9004	453.5294	5.25
B7	1698.601	1701.9693	0.19
B8	791.3677	799.649	1.04
B10	164.9191	154.3107	6.43
B11	3817.71	3806.8961	0.28
Average error			3.12

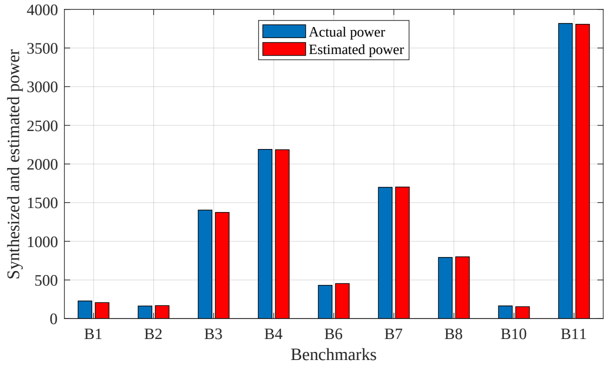


Fig. 5: Comparison of synthesized and estimated power.

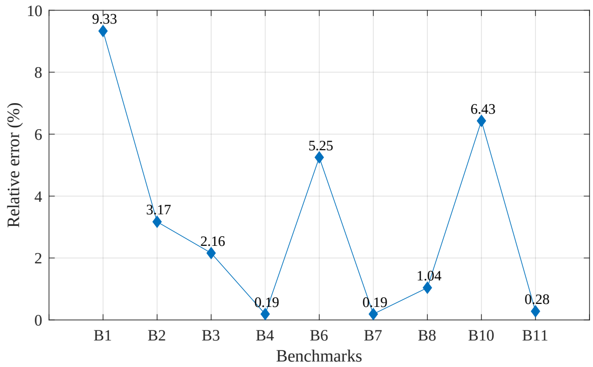


Fig. 6: Observed error in the estimation.

The area estimation comparison with classical method is shown in Fig. 7. The relative error percentage in the area estimation is shown in Fig. 8.

The run-times are faster than the synthesis runtimes. The area and power estimation time of the proposed model compiled on an Intel core i3 processor with clock

Tab. 2: Comparison of Estimated area and Synthesis area for ITC99 benchmarks.

Benchmarks [23]	Synthesis area	Estimated area	Relative error
B1	83	82.0783	1.11
B2	58	53.6867	7.43
B3	382	384.5271	0.66
B4	982	968.8896	1.33
B6	130	131.3214	1.01
B7	776	777.9353	0.24
B8	314	312.1492	0.58
B10	311	294.6238	5.26
B11	793	794.5067	0.19
Average error			1.98

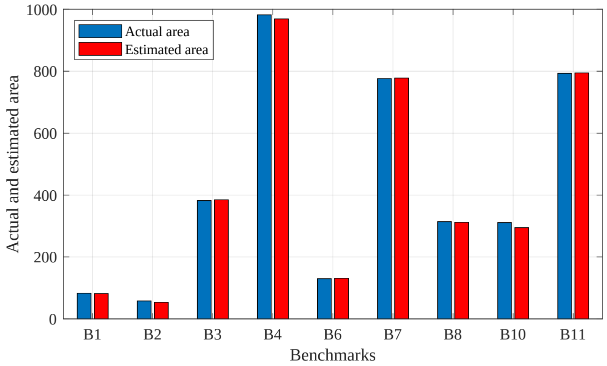


Fig. 7: Comparison of synthesized and estimated area.

frequency of 3.30 GHz, was 5 seconds, while the results obtained form the synopsys design compiler for the same machine took 5 minutes to estimate the power and area for ITC99 benchmark applications. The proposed method is 12 times faster than the conventional method. Moreover, the proposed method is better than other works. The comparison is shown in Tab. 3 and Tab. 4.

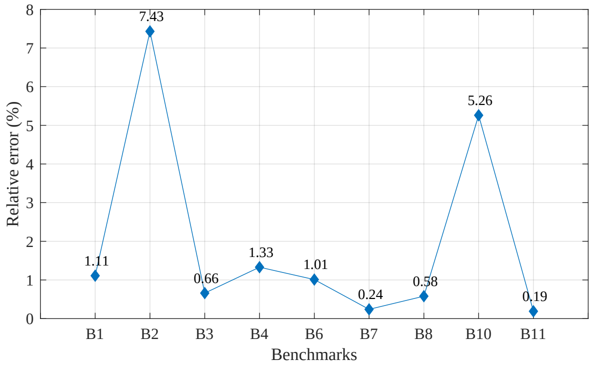


Fig. 8: Observed error in the estimation.

Tab. 3: Comparison with other works for power estimation.

Sr. No.	Approaches	Average error %
1.	[1]	10 %
2.	[24]	6 %
3.	Proposed method	3.12 %

Tab. 4: Comparison with other works for area estimation.

Sr. No.	Approaches	Average error %
1.	[25]	3.2 % to 4 %
2.	[16]	6.1 %
3.	Proposed method	1.98 %

5. Conclusion

In this paper high-level area and power estimation method for the ASIC based designs have been presented. In terms of timing, the estimation speed is more than an order of magnitude faster of the commercial tools i.e. Synopsys Design Compiler. The estimation error is in the range varies from 0.19 % to 7.43 % and from 0.19 % to 9.33 % for area and power, respectively. In addition, different works have targeted either power or area only, while the proposed method has targeted both.

The generic approach for area and power estimation, presented in this paper may be applicable to many other estimation problems such as estimation of cost, execution time etc. In addition, the proposed model can be conveniently used for rapid design space exploration.

Author Contributions

A.N.T. have contributed to conceptualization, methodology, software, writing - original draft preparation, writing - review, proofreading and editing. A.R. have contributed to conceptualization, supervision and project administration.

References

[1] LEE, D., L. K. JOHN and A. GERSTLAUER. Dynamic power and performance back-annotation for fast and accurate functional hardware simulation. In: *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Grenoble: IEEE, 2015, pp. 1126–1131. ISBN 978-3-9815370-5-5.

[2] MEEUWS, R. *A Quantitative model for Hardware/Software Partitioning*. Delft, 2007. Master's thesis. Delft University of Technology. Supervisor Koen Bertels, Ph.D.

[3] SRINIVASAN, V., S. GOVINDARAJAN and R. VEMURI. Fine-grained and coarse-grained behavioral partitioning with effective utilization of memory and design space exploration for multi-FPGA architectures. *IEEE Transactions on Very*

Large Scale Integration (VLSI) Systems. 2001, vol. 9, iss. 1, pp. 140–158. ISSN 1557-9999. DOI: 10.1109/92.920829.

[4] VAHID, F. and D. D. GAJSKI. Incremental hardware estimation during hardware/software functional partitioning. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 1995, vol. 3, iss. 3, pp. 459–464. ISSN 1557-9999. DOI: 10.1109/92.407006.

[5] YAN, L., T. SRIKANTHAN and N. GANG. Area and delay estimation for FPGA implementation of coarse-grained reconfigurable architectures. *ACM SIGPLAN Notices*. 2006, vol. 41, iss. 7, pp. 182–188. ISSN 1558-1160. DOI: 10.1145/1159974.1134677.

[6] NAYAK, A., M. HALDAR, A. CHOUDHARY and P. BANERJEE. Accurate area and delay estimators for FPGAs. In: *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*. Paris: IEEE, 2002, pp. 862–869. ISBN 978-0-7695-1471-0. DOI: 10.1109/DATE.2002.998400.

[7] KULKARNI, D., W. A. NAJJAR, R. RINKER and F. J. KURDAHI. Fast area estimation to support compiler optimizations in FPGA-based reconfigurable systems. In: *Proceedings. 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*. Napa: IEEE, 2002, pp. 239–247. ISBN 978-0-7695-1801-5. DOI: 10.1109/FPGA.2002.1106678.

[8] XU, M., and F. J. KURDAHI. Accurate prediction of quality metrics for logic level designs targeted toward lookup-table-based FPGAs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 1999, vol. 7, iss. 4, pp. 411–418. ISSN 1557-9999. DOI: 10.1109/92.805748.

[9] NEMANI, M. and F. N. NAJM. High-level area and power estimation for VLSI circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1999, vol. 18, iss. 6, pp. 697–713. ISSN 1937-4151. DOI: 10.1109/43.766722.

[10] JIANG, T., X. TANG and P. BANERJEE. Macro-models for high level area and power estimation on FPGAs. In: *Proceedins of the 14th ACM Great Lakes symposium on VLSI (GLSVLSI '04)*. Boston: ACM Press, 2004, pp. 340–343. ISBN 978-1-58113-853-5. DOI: 10.1145/988952.988992.

[11] KAISER, S., I. MATERIC and R. SAADE. ESL solutions for low power design. In: *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. San Jose: IEEE, 2010, pp. 340–343. ISBN 978-1-4244-8194-1. DOI: 10.1109/ICCAD.2010.5653615.

[12] BHARGAVA, M., L. BHARGAVA and V. SAHULA. A study of low power design techniques for ASIPs. In: *IETE Annual Zonal seminar on Electronic Design Automation: Issue & Challenges*. Jaipur: 2003.

[13] MATHE, S. E. and L. BOPPANA. Low-Power and Low-Hardware Bit-Parallel Polynomial Basis Systolic Multiplier over GF(2^m) for Irreducible Polynomials. *ETRI Journal*. 2017, vol. 39, iss. 4, pp. 570–581. ISSN 1225-6463. DOI: 10.4218/etrij.17.0116.0770.

[14] GIAMMARINI, M., M. CONTI and S. ORCIONI. System-level energy estimation with Powersim. In: *2011 18th IEEE International Conference on Electronics, Circuits, and Systems*. Beirut: IEEE, 2011, pp. 723–726. ISBN 978-1-4577-1846-5. DOI: 10.1109/ICECS.2011.6122376.

[15] DENG, C., L. LIU, Y. LIU, S. YIN and S. WEI. PMCC: Fast and Accurate System-Level Power Modeling for Processors on Heterogeneous SoC. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2017, vol. 64, iss. 5, pp. 540–544. ISSN 1558-3791. DOI: 10.1109/TCSII.2016.2615930.

[16] MILDER, P. A., M. AHMAD, J. C. HOE and M. PUSCHEL. Fast and accurate resource estimation of automatically generated custom DFT IP cores. In: *Proceedings of the 2006 ACM/SIGDA 14th International Symposium on Field Programmable Gate Arrays (FPGA '06)*. Monterey: ACM Press, 2006, pp. 211–220. ISBN 978-1-59593-292-1. DOI: 10.1145/1117201.1117232.

[17] DENG, L., K. SOBTI and C. CHAKRABARTI. Accurate models for estimating area and power of FPGA implementations. In: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. Las Vegas: IEEE, 2008, pp. 1417–1420. ISBN 978-1-4244-1483-3. DOI: 10.1109/ICASSP.2008.4517885.

[18] KUMAR, R. and C. P. RAVIKUMAR. Leakage power estimation for deep submicron circuits in an ASIC design environment. In: *Proceedings of ASP-DAC/VLSI Design 2002. 7th Asia and South Pacific Design Automation Conference and 15h International Conference on VLSI Design*. Bangalore: IEEE, 2002, pp. 45–50. ISBN 978-0-7695-1441-3. DOI: 10.1109/ASPDAC.2002.994883.

[19] WANG, L., C. ZHUO and P. ZHOU. Run-time demand estimation and modulation of on-chip decaps at system level for leakage power reduction in multicore chips. *Integration*. 2019, vol. 65, iss. 1, pp. 322–330. ISSN 0167-9260. DOI: 10.1016/j.vlsi.2018.01.009.

[20] ZHOU, Y., H. REN, Y. ZHANG, B. KELLER, B. KHAILANY and Z. ZHANG. PRIMAL: Power Inference using Machine Learning. In: *Proceedings of the 56th Annual Design Automation Conference 2019 (DAC '19)*. Las Vegas: ACM Press, 2019, pp. 1–6. ISBN 978-1-4503-6725-7. DOI: 10.1145/3316781.3317884.

[21] Design Compiler Graphical. In: *Synopsys* [online]. Available at: `https://www.synopsys.com/implementation-and-signof\protect\unhbox\voidb@x\hbox{}f/rtl-synthesis-test/design-compiler-graphical.html`.

[22] MATLAB–Neural Network Toolbox. In: *MathWorks* [online]. Available at: `https://www.mathworks.com`.

[23] CORNO, F., M. S. REORDA and G. SQUILLERO. RT-level ITC'99 benchmarks and first ATPG results. *IEEE Design & Test of Computers*. 2000, vol. 17, iss. 3, pp. 44–53. ISSN 1558-1918. DOI: 10.1109/54.867894.

[24] AHUJA, S., D. A. MATHAIKUTTY, G. SINGH, J. STETZER, S. K. SHUKLA and A. DINGANKAR. Power estimation methodology for a high-level synthesis framework. In: *2009 10th International Symposium on Quality Electronic Design*. San Jose: IEEE, 2009, pp. 541–546. ISBN 978-1-4244-2952-3. DOI: 10.1109/ISQED.2009.4810352.

[25] EEROLA, V. and J. NURMI. High-level parameterizable area estimation modeling for ASIC designs. *Integration*. 2014, vol. 47, iss. 4, pp. 461–475. ISSN 0167-9260. DOI: 10.1016/j.vlsi.2014.01.002.

About Authors

Abhishek Narayan TRIPATHI received the B.E. degree in Electronics and Communication engineering from the Rajiv Gandhi Proudyogiki Vishwavidyalaya University, Bhopal, India, in 2008, and the M.Tech degree in microelectronics and embedded technology from the Jaypee Institute of Information Technology, Noida, India, in 2012. He has received the Ph.D. degree from Maulana Azad National Institute of Technology Bhopal in 2020. Currently, He is working as an assistant professor in electronics and communication engineering with the National Institute of Technology Raipur, Chhattisgarh, India. His research interests include system-level design, VLSI Design design space exploration.

Arvind RAJAWAT received his Bachelor of Engineering in Electronics and Communication Engineering from Government Engineering College Ujjain (Madhya Pradesh, India) in 1989, Master of Engineering in Computer Engineering from Shri Govindram Seksaria Institute of Technology and Science (SGSITS) Indore (Madhya Pradesh, India) in 1991 and Ph.D. from Maulana Azad National Institute of Technology Bhopal (Madhya Pradesh, India) in 2009. He is currently working as professor of Electronics and Communication Engineering with the department of Electronics and Communication Engineering at Maulana Azad National Institute of Technology Bhopal (Madhya Pradesh, India). His areas of interest are hardware software co-design, embedded sysem design, digital system design.